

EXHIBIT 21

Contains Highly Confidential AEO and Source Code Materials

IN THE UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

GOOGLE LLC,

Plaintiff

v.

SONOS, INC.,

Defendant.

CASE NO. 3:20-cv-06754-WHA

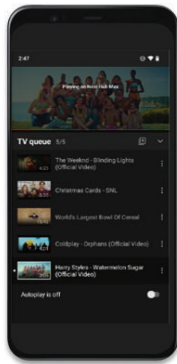
Related to CASE NO. 3:21-cv-07559-WHA

**REBUTTAL EXPERT REPORT OF SAMRAT BHATTACHARJEE REGARDING NON-
INFRINGEMENT OF CLAIM 13 OF U.S. PATENT NO. 9,967,615 AND OTHER ISSUES**

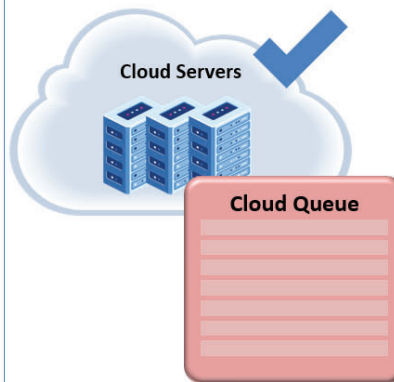
HIGHLY CONFIDENTIAL AEO AND SOURCE CODE MATERIALS

Contains Highly Confidential AEO and Source Code Materials

Users Can Create Playlist With Hundreds Of Items



Playlist Is Stored In A Cloud Queue



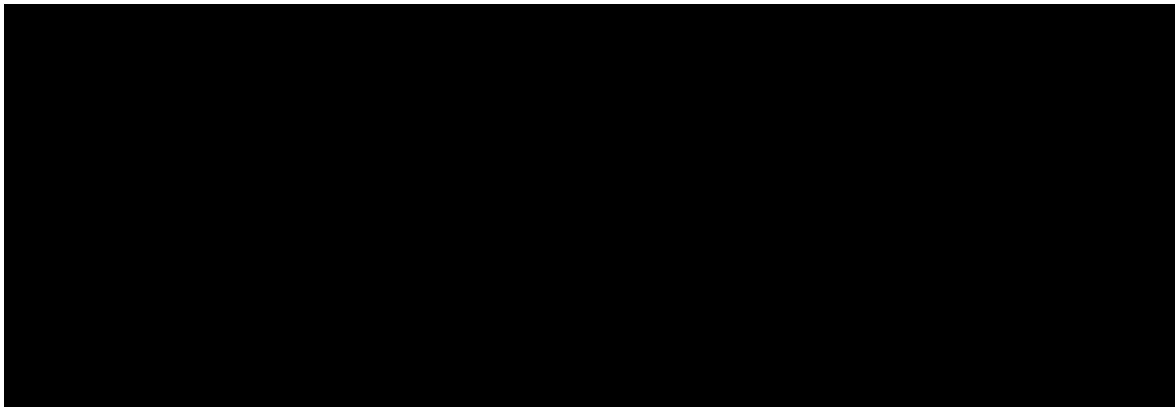
Receiver Fetches Identity Of The Next Item In The Cloud Queue One-By-One



3. Fetching Videos Or Songs From Google's Content Delivery Network

94. The accused YouTube playback devices retrieve songs and videos from Google's Content Delivery Network ("CDN"). Below I provide an overview of Google's CDN and the retrieval of YouTube songs and videos.

95. Google's Content Delivery Network (CDN) is sometimes referred to as [REDACTED]. The CDN deploys cache servers across the globe.



GOOG-SONOSNDCA-00115814 ([REDACTED]) at slide 3.

96. [REDACTED]

[REDACTED]

Contains Highly Confidential AEO and Source Code Materials

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]. GOOG-SONOSNDCA-00115893

(“[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED], GOOG-SONOSNDCA-00115814 [REDACTED]

[REDACTED]

97. Each YouTube song or video includes an identifier (called a videoId) and the same song or video may exist on many different CDN servers at a given time. A videoId does not provide any information regarding the location of a song or video on a Bandid CDN server. Indeed, the process for locating the multimedia content involves a number of steps. For instance, before an alleged playback device can even make a request for a song or video, it must first make a request to a “Player Service” (called a GetPlayer request). The Player Service [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] GOOG-SONOSNDCA-00115893 (“[REDACTED]

Contains Highly Confidential AEO and Source Code Materials

[REDACTED]

[REDACTED]

[REDACTED]

GOOG-SONOSNDCA-00115814 ([REDACTED]) at slide 8 [REDACTED]

[REDACTED]

[REDACTED]).

98. For instance, a [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] GOOG-SONOSNDCA-00115893. Notably, the Mapping Service uses factors, such as the client IP address and network conditions, that are entirely unrelated to the videoId, which reinforces my view that the videoId is not “translated” into the URL as Dr. Schmidt suggests. And because the Mapping Service uses dynamic factors, the same videoId can result in different URLs for each user or for the same user each time the video is played back.

Contains Highly Confidential AEO and Source Code Materials

[REDACTED] See also GOOG-SONOSNDCA-00115814 ([REDACTED]) at slide 2.

B. Overview Of The Google Play Music Application

101. The Google Play Music (“GPM”) application was a music application that provided music streaming services to users. I understand GPM was discontinued in 2020.

102. By downloading and installing the GPM application on a mobile device that supports the application, users could view available music, add music to a playlist, and playback music on their mobile device. The GPM application also permitted users to send playback of a playlist from the mobile device through a server to a receiver device, such as a speaker or TV through Google’s “casting” feature. I refer to the GPM application, receiver device, and the remainder of the system as a “GPM System.”

103. The cast feature was discussed above with respect to the accused YouTube system. The differences between casting in YouTube and GPM are not material to my opinions. Thus, I incorporate my discussion of casting in the YouTube applications section above.

104. Playback of music on a receiver device was managed by Google’s Cloud Queue API. In the GPM system, the queue was stored on a Cloud Queue (or CQ) Server, and a receiver device played back items from the cloud queue. I discuss the Cloud Queue API in further detail below.

1. Development Of Google Play Music And Cloud Queue API

105. On May 20, 2010, at Google I/O 2010, Google “showed off a new section of the [Android] Market - Music. Yes, an iTunes competitor on the web from Google.”¹⁵ With this new Music offering a user could “go to the [Android] Market on the web, find a song [the user] like[d],

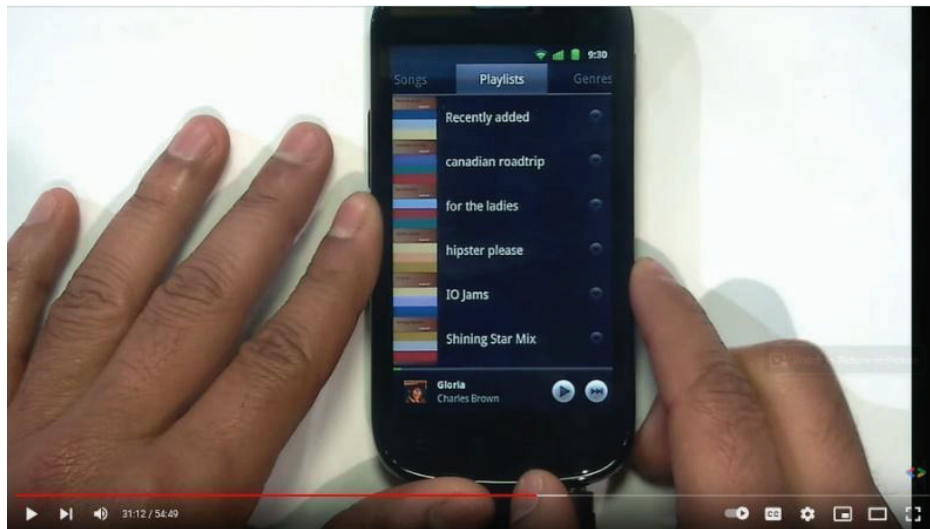
¹⁵ <https://techcrunch.com/2010/05/20/um-did-google-just-quietly-launch-a-web-based-itunes-competitor-yep/>

Contains Highly Confidential AEO and Source Code Materials

click the download button, and just like with apps, the song starts to download on your Android devices.” The Music offering was also described as “iTunes, over the web, with auto-syncing.”

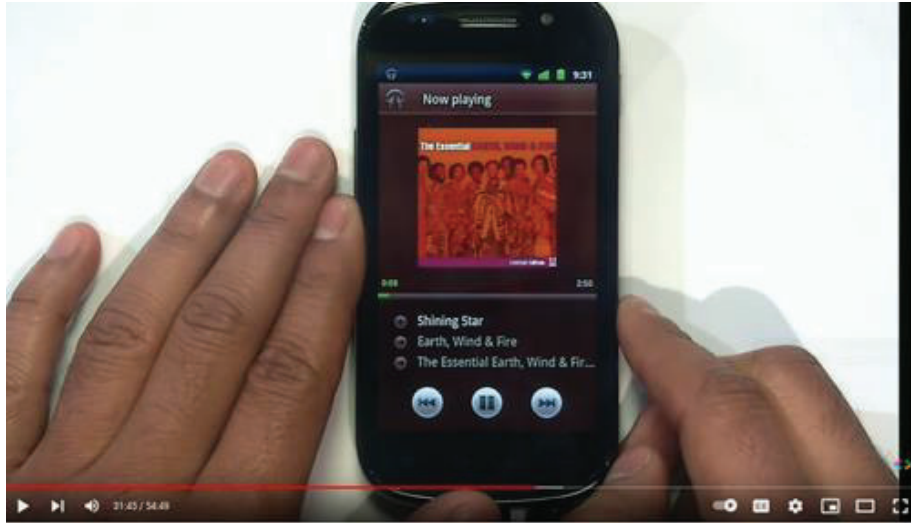
106. The next year, at the May 2011 Google I/O, Google officially announced “[t]wo new cloud-based players . . . [which] will allow users to stream music and movies.”¹⁶ This included a beta version of Google’s Music application, called “Music Beta.” Music Beta was referred to as a “digital music locker,” and which “allow[ed] users to upload up to 20,000 songs from their computer . . . to the internet, which they can then listen to on the move.” GOOG-SONOSNDCA-00116488.

107. For example, the following video provides a demonstration of the Music Beta application on a user’s phone that was demonstrated at the May 2011 I/O, including a user creating a playlist and playing back songs on the user’s device:



¹⁶ <https://www.theguardian.com/technology/2011/may/10/google-launches-music-beta-film-player>

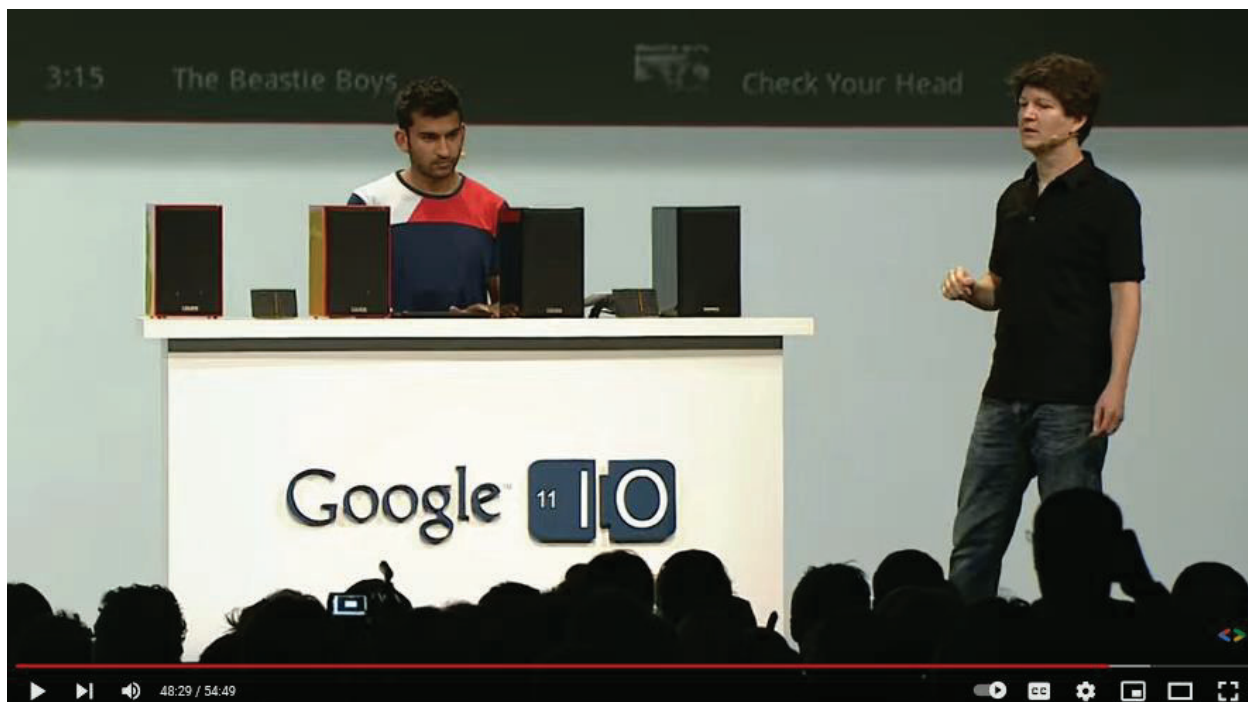
Contains Highly Confidential AEO and Source Code Materials



<https://youtu.be/OxzucwjFEs?t=2808>, Google I/O 2011, Keynote Day 1, uploaded May 10, 2011 (GOOG-SONOS-WDTX-INV-00015090), *see also* <https://www.youtube.com/watch?v=9ZlgcuG3sZc> (uploaded on 5/10/2011), GOOG-SONOS-WDTX-INV-00015092.

108. As I showed in my opening report, the May 2011 Google/IO further demonstrated that the Music Beta application could be used to cast playback to a Project Tungsten speaker using a device-picker (e.g., choosing StageLeft or StageRight).

Contains Highly Confidential AEO and Source Code Materials



<https://youtu.be/OxzucwjFEs?t=2808>, Google I/O 2011, Keynote Day 1, uploaded May 10, 2011 (GOOG-SONOS-WDTX-INV-00015090), *see* *also* <https://www.youtube.com/watch?v=9ZlgbuG3sZc> (uploaded on 5/10/2011). As explained at the

Contains Highly Confidential AEO and Source Code Materials

I/O: “You can imagine that this device is playing music in, say, your living room. If we take a look at the tablet Anand is using, we can see he’s in the music app. But there’s a new feature to select an output device. The tablet can direct music to one or more Tungsten boxes like the one we have here. Anand, why don’t you start music on both of them. You can enjoy your music synchronized throughout your house, all streaming through Music Beta. When Anand tapped on those buttons, the music stream was sent transparently from one box to another. Since the boxes are running Android, they just pull the music directly from the music library in the cloud.” *Id.* at 46:50-49:00.

109. By December of 2011 Google had developed a Music2 application that was used by the Tungsten/Nexus Q prior art system, as I showed in my opening Report ([REDACTED]).

110. I understand that this Music2 application is what eventually became the Google Play Music application.¹⁷ For example, Google engineer Debajit Gosh testified that the Music app was an early name for Google Play Music:

Q. And what names did the app have?

A. One of the early names of the app was something like Music Beta by Google. I -- I can’t remember the exact name of the app. Like I said, the app name changed a number of times over the years.

Q. Do you remember any other names for the app?

A. I do remember some other names.

Q. And what names were those?

A. Another name I remember is Google Music. 09:37:31

Q. Were there any other names?

A. There were some other names for the app.

Q. And what were those?

A. Another name for the app, I believe, was Google Play Music.

Gosh Tr. at 23:5-20.

¹⁷ Calls with Eugene Koh, John Grossman and Jeremy Ellington.

Contains Highly Confidential AEO and Source Code Materials

292. Dr. Schmidt opines that “Google’s source code [REDACTED]
[REDACTED].” Schmidt Rpt., 76. In particular, Dr. Schmidt cites to [REDACTED]
[REDACTED]
[REDACTED]
[REDACTED] Moreover, a person of skill in the art would understand that the source code is what actually dictates the operation and functionalities of the accused products, regardless of what the comments in the file say. [REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

33

293. Dr. Schmidt also references an internal correspondence which stated [REDACTED]
[REDACTED] Schmidt Rpt., 77 (citing GOOG-SONOS-NDCA-00075833); *see also id.*, ¶¶285, 452. This statement is made by Chris Patnoe regarding Google Play Music. I understand that Mr. Patnoe is not an engineer and does not have a technical degree, and that his role at Google at the time of this email was that of a program manager. Patnoe Depo Tr. at 11:13-12:2 (“Q. As a program manager, what were your

³³ I also spoke with Google engineer, Umesh Patil, who confirmed that the receiver does not store a playback queue. Rather, the comment, although not clearly worded, was intended to capture the fact that when a change is made to the playback queue on the sender device, the cloud queue is notified of the change, a new version number is assigned to the queue, and the receiver refreshes its view of the cloud queue. This is consistent with what I have seen in the source code, which actually implements the functionality, as opposed to the comments which do not.

Contains Highly Confidential AEO and Source Code Materials

roles and responsibilities, generally? A. Making sure that people were aware of what was happening, delivering products at a high quality, and doing all logistical and communication to work to make that happen. Q. Do you have any university degrees, Mr. Patnoe? A. Music. Music. U.C Berkely. University of California Berkeley.... Q. Do you have any other degrees besides that one? A. No.”). In other words, this appears to be Mr. Patnoe, a non-technical person, writing a non-technical email in response to a reported problem with casting to Chromecast Audio. Further, it is unclear what Mr. Patnoe was referring to when making the statement [REDACTED] [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

6. GPM Stores the Playback Queue In A Cloud Queue

294. Unlike the alleged invention of Claim 13 of the ‘615 patent, the accused GPM application stores does not store the playback queue locally on the receiver device-the playback queue is instead stored in a cloud queue.

295. For example, Google’s documents consistently explain that GPM receivers playback a cloud queue on a “Cloud Queue” server:

[REDACTED]

GOOG-SONOSWDTX-00043627 at 28; *see also* GOOG-SONOSWDTX-00053379 [REDACTED]
[REDACTED]); GOOG-SONOSWDTX-00043467 [REDACTED]). While

Contains Highly Confidential AEO and Source Code Materials

not even have in its possession a Cloud Queue API until years after the alleged invention date further supports my opinion that the ‘615 patent’s “local playback queue” was not intended to cover the Cloud Queue API that Sonos is now accusing.

297. This conclusion is further supported by other statements from Sonos witnesses, including the inventor of the ‘615 patent, Tad Coburn. For instance, as I discussed in my opening report, Google stored the playback queue locally on its receiver devices in its prior art products. Sonos also stored the playback queue on its receiver devices at the time of the alleged invention. However, in 2013 Google worked with Sonos to move the playback queue to the cloud. Notably, when Google informed Sonos that it was planning to move to a cloud queue, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

298. It is also notable that Sonos’s witnesses testified [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] *See e.g., Adam Graham Dep. Tr. at 42:11-44:7.* This is yet further evidence that Sonos did not consider its “local playback queue” patent to cover a cloud queue at the time of the collaboration.

299. Dr. Schmidt opines that the “local playback queue” is an `itemWindowResponse`. Schmidt Rpt., 382-385, 438, 442-444. I disagree that an `itemWindowResponse` is a “playback queue.”

Contains Highly Confidential AEO and Source Code Materials

considerations. It is also my understanding that Sonos may submit an expert report corresponding to this report. I reserve the right to rebut any positions taken in that report.

394. I, Samrat Bhattacharjee, declare under penalty of perjury under the laws of the United States that the foregoing is true and correct.

DATED: July 27, 2022



Dr. Samrat Bhattacharjee, PhD.